Concordia University
Department of Computer Science
and Software Engineering

Advanced program design with C++
COMP 345 --- Fall 2017

Team project assignment #4

Deadline:	November 23rd, 2017
Evaluation:	5% of final mark
Late submission:	not accepted
Teams:	this is a team assignment

Problem statement

This is a team assignment. It is divided into distinct parts. Each part is about the development of a part of the topic presented as the team project. Even though it is about the development of a part of your team project, each assignment is to be developed/presented/tested separately. The description of each part describes what are the features that the part should implement, and what you should demonstrate. Note that the following descriptions describe the baseline of the assignment, and are related to the project description. See the course web page for a full description of the team project, as well as links to the details of the game rules to be implemented.

Part 1: Game Statistics Observer Decorator

(Extension of Part 3 of Assignment 3) Using the Observer design pattern, implement a view that displays some useful statistics about the game as it is being played. This should dynamically be updated as various aspect of the game state changes and be visible at all times during game play. Using the Decorator pattern, provide different decorators that add more information to be displayed on the Basic (undecorated) Game Statistics Observer:

1. Basic (undecorated) Game Statistics Observer: Display the turn number, update the view every time a new turn starts.
2. Player Domination Observer Decorator: Display the percentage of countries owned by each player, update the view when any player conquers a country.
3. Player Hands Observer Decorator: Display the cards owned by every player, update the view when any player's hand is changing.
4. Continent Control Observer Decorator: Display what player controls each continent, update the view when any continent becomes controlled by a player, or when a player loses control of a continent.

You must be able to demonstrate that the Basic (undecorated) Game Statistics Observer is available at all times during play and updates the view when the turn number increases. Before the beginning of every turn, the user should be given the opportunity to add one or more Decorators (2, 3, or 4) above, which should result in the Observer to show more information and be notified as instructed above for each Decorator. Each Decorator (2,3,4) should not be used more than once. The user should also be allowed to remove Decorators. The user should be allowed to specify that it does not want to add/remove Decorators in the future, after which the user shall not be prompted anymore at the beginning of every turn.

Part 2: New Strategies

(Extension of Part1 of Assignment 3) Add 2 more computer strategies (you should then have a total of 4 different computer strategies). These strategies must be designed so that they have a chance to win the game (e.g. the "benevolent" strategy in assignment 3 cannot win a game).

- A Random Computer Player Strategy that reinforces random a random country, attacks a random number of times a random country, and fortifies a random country, all following the standard rules for each phase,
- A Cheater Computer Player Strategy whose reinforce() method doubles the number of armies on all its countries, whose attack() method automatically conquers all the neighbors of all its countries, and whose fortify() method doubles the number of armies on its countries that have neighbors that belong to other players.

Part 3: Tournament

A tournament starts with the user choosing M = 1 to 5 different maps, P = 2 to 4 different computer players strategies, G = 1 to 5 games to be played on each map, D = 10 to 50 maximum number of turns for each game. A tournament is then automatically played by playing G games on each of the M different maps between the chosen computer player strategies. In order to minimize run completion time, each game should be declared a draw after D turns. Once started, the tournament plays all the games automatically without user interaction. At the end of the tournament, a report of the results should be displayed, e.g.

M: Map1, Map2, Map3
P: Aggressive, Benevolent, Random, Cheater.
G: 4
D: 30

	Game 1	Game 2	Game 3	Game 4
Map 1	Aggressive	Random	Cheater	Cheater
Map 2	Cheater	Draw	Cheater	Aggressive
Map 3	Cheater	Aggressive	Cheater	Draw

Assignment submission requirements and procedure

You are expected to submit a group of C++ files implementing a solution to each of the separate problems stated above (Part 1, 2, 3). Your code must include a *driver* (i.e. a `main` function) for each part that allows the marker to observe the execution of each part during the lab demonstration. Each driver should simply create the components described above and demonstrate that they behave as mentioned above.

You have to submit your assignment before midnight on the due date using the ENCS Electronic Assignment Submission system under the category "programming assignment 4". Late assignments are not accepted. The file submitted must be a .zip file containing all your code. You are allowed to use any C++ programming environment as long as you can demonstrate your assignment in the labs.

Evaluation Criteria

Knowledge/correctness of game rules:	2 pts (indicator 4.1)
Compliance of solution with stated problem (see description above):	12 pts (indicator 4.4)
Modularity/simplicity/clarity of the solution:	2 pts (indicator 4.3)
Proper use of language/tools/libraries:	2 pts (indicator 5.1)
Code readability: naming conventions, clarity of code, use of comments:	2 pts (indicator 7.3)
Total	**20 pts (indicator 6.4)**